Linear Programming Detection and Decoding for MIMO Systems

Tao Cui, Tracey Ho
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA, USA 91125
Email: {taocui, tho}@caltech.edu

Chintha Tellambura
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB, Canada T6G 2V4
Email: chintha@ece.ualberta.ca

***Abstract*— [1]We develop an efficient linear programming detector (LPD) for multiple-input multiple-output (MIMO) systems. Instead of using the usual l_2 norm, our proposed LPD uses the l_1 norm as the detection metric, resulting in a mixed-integer linear program (MILP). Two branch-and-bound algorithms are proposed to solve the MILP. The solution of the MILP achieves the same full diversity order as the maximum likelihood detector. The MILP is further relaxed to a linear program (LP), which can be readily solved using the standard simplex method. We show that in some cases the solution of the LP is guaranteed to be that of the MILP. The LPD is also extended to the joint detection and decoding of linear block coded MIMO systems. Our LPD can be immediately implemented using mature circuits design for the simplex algorithm.**

I. Introduction

Multiple-input multiple-output (MIMO) systems, employing multiple antennas at both the transmitter and the receiver, can achieve remarkably high spectral efficiencies in rich scattering environments. As a result, the design of high data-rate MIMO communication systems has attracted a significant interest. A prime example is the BLAST (Bell Laboratories layered space time) architecture [1], which has been proposed to exploit the potentially enormous MIMO link capacity.

In spatial multiplexing systems, a fundamental receiver function is the detection of transmitted data symbols. The optimal maximum likelihood detector (MLD) minimizes the error probability. However, the complexity of the MLD grows exponentially with the number of transmit antennas, making it computationally prohibitive in many cases. Consequently, various efficient but suboptimal receivers are popular. In [1], the V-BLAST detector with optimal ordering, nulling and interference cancellation is proposed. However, it can only achieve a diversity order one. In [2], sphere decoding (SD), an algorithm for the MLD, is proposed for attaining low complexity in high SNR. Although the VLSI implementation of SD has been reported [3], the variation of its time complexity can be high, leading to undesirable variable detection delays. Alternative detectors with constant time complexity are thus desirable. However, all these detectors are based on l_2 norm.

In this paper, we consider a linear programming (LP) formulation of the MIMO detection problem. We observe that the bottleneck of the former detectors [1], [2] is the use of l_2 norm as the metric resulting in complicated integer quadratic programming [2] or involving in matrix decomposition [1], [2]. In [4], [5], LP is used for decoding of linear codes. As in [4], [5], we use l_1 norm in our linear programming detector (LPD), which results in a mixed-integer linear program (MILP). Two branch-and-bound (BnB) algorithms are proposed to solve the MILP. We prove that the solution of the MILP achieves the same full diversity order as the maximum likelihood detector. The MILP is further relaxed to a LP, which can be readily solved using standard simplex methods. We show that in some cases the solution of LP is guaranteed to be that of the MILP. We also consider the LPD for the joint detection and decoding of linear block coded MIMO systems. The LP decoding for linear block codes over additive white Gaussian noise (AWGN) channel in [6] is adopted into the LPD for the uncoded MIMO systems. The major advantage of our LPD is that it can be immediately implemented using mature circuits design for the simplex algorithm [7].

Notation: $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex number sets. $\Re\{x\}$ and $\Im\{x\}$ denote the real part and imaginary part of x, respectively. $\|(\cdot)\|_2$ and $\|(\cdot)\|_1$ are the l_2-norm and l_1-norm of $(\cdot)$. A circularly complex Gaussian variable with mean μ and variance σ^2 is denoted by $z \sim \mathcal{CN}(\mu, \sigma^2)$. $\jmath = \sqrt{-1}$.

II. System model

We consider a spatial multiplexing MIMO system with n transmit antennas and m receive antennas. Source data are mapped into complex symbols from a finite constellation $\tilde{\mathcal{Q}}$. We assume a rich scattering memoryless channel. The received signals may be written as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}} \tag{1}$$

where $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_n]^T$, $\tilde{x}_i \in \tilde{\mathcal{Q}}$ is the transmitted signal vector, $\tilde{\mathbf{y}} = [\tilde{y}_1, \ldots, \tilde{y}_m]^T$, $\tilde{y}_i \in \mathbb{C}$ is the received signal vector, $\tilde{\mathbf{H}} = [\tilde{h}_{i,j}] \in \mathbb{C}^{m \times n}$ is the channel matrix, and $\tilde{\mathbf{n}} = [\tilde{n}_1, \ldots, \tilde{n}_m]^T \in \mathbb{C}^m$ is an additive white Gaussian noise (AWGN) vector. The elements of $\tilde{\mathbf{H}}$ are identically independent distributed (i.i.d.) complex Gaussian, $\tilde{h}_{i,j} \sim \mathcal{CN}(0, 1)$. The components of $\tilde{\mathbf{n}}$ are i.i.d. and $\tilde{n}_i \sim \mathcal{CN}(0, \sigma_n^2)$. We assume that the channel is perfectly known to the receiver. But we do not assume $n \leq m$ because without modification, our LPD can be directly used to solve the rank deficient system with $n > m$. Note this model (1) is not restricted to MIMO systems. It models any linear, synchronous and

[1]This work was supported in part by Caltech's Lee Center for Advanced Networking.

memoryless channels with crosstalk. For brevity, we restrict our considerations to MIMO.

The MLD that minimizes the average error probability is

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}^n} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{x}}\|_2^2. \tag{2}$$

Due to the discrete nature of $\mathcal{Q}$, (2) is a NP-hard problem and exhaustive search for $\hat{\mathbf{x}}$ has a complexity exponential in n.

III. Linear programming detection

The l_1 norm is widely used for data analysis and parameter estimation [8], because the cost function using l_1 norm metric is not only robust to outliers, but also computationally tractable.

A. MILP

In this paper, we suggest detecting data symbols using l_1 norm metric or

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}^n} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{x}}\|_1. \tag{3}$$

In fact, if we assume the noise vector $\tilde{\mathbf{n}}$ follows the Laplace distribution instead of the normal distribution, the MLD of the data symbols is given by (3). Since l_1 norm metric is robust to outliers, (3) may be robust to the impulsive noise due to the impulsive nature of man-made electromagnetic interference and a great deal of natural noise.

Note that (3) involves complex operations. We transform (1) to a real system as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{4}$$

where

$$\mathbf{y} = \begin{bmatrix} \Re\{\tilde{\mathbf{y}}\} \\ \Im\{\tilde{\mathbf{y}}\} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \Re\{\tilde{\mathbf{x}}\} \\ \Im\{\tilde{\mathbf{x}}\} \end{bmatrix}, \mathbf{n} = \begin{bmatrix} \Re\{\tilde{\mathbf{n}}\} \\ \Im\{\tilde{\mathbf{n}}\} \end{bmatrix} \tag{5}$$

and

$$\mathbf{H} = \begin{bmatrix} \Re\{\tilde{\mathbf{H}}\} & -\Im\{\tilde{\mathbf{y}}\} \\ \Im\{\tilde{\mathbf{y}}\} & \Re\{\tilde{\mathbf{y}}\} \end{bmatrix}. \tag{6}$$

We assume that $\tilde{\mathcal{Q}}$ is a decouplable constellation, i.e., squared quadrature amplitude modulation (QAM). $\Re\{x_i\}$ and $\Im\{x_i\}$ belong to the same constellation $\mathcal{Q}$. For example, for 4-QAM, $\mathcal{Q} = \{-1, 1\}$.

To detect data symbols $\mathbf{x}$ in (4), we consider solving the following l_1-minimization problem

$$(\mathrm{P}_1) \quad \min_{\mathbf{x} \in \mathcal{Q}^{2n}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_1 \tag{7}$$

Since all the variables in (7) are real, by introducing slack variables t_i, $i = 1, \ldots, 2n$, (P_1) can be classically reformulated as an MILP. (P_1) is equivalent to

$$\begin{array}{lll} (\mathrm{MILP}_1) & \min & \sum_{i=1}^{2n} t_i \\ & \text{s.t.} & \mathbf{H}\mathbf{x} - \mathbf{t} \le \mathbf{y} \\ & & -\mathbf{H}\mathbf{x} - \mathbf{t} \le -\mathbf{y} \\ & & x_i \in \mathcal{Q},\ i = 1, \ldots, 2n, \end{array} \tag{8}$$

where $\mathbf{t} = [t_1, \ldots, t_{2n}] \in \mathbb{R}^{2n}$ and $\mathbf{x} \in \mathcal{Q}^{2n}$ are the optimization variables. The generalized vector inequality $\mathbf{x} \le \mathbf{y}$ means that $x_i \le y_i$ for every coordinate i. If $\mathcal{Q}$ is a subset of the integer set $\mathbb{Z}$, (8) is an MILP. Note that the elements of $\mathcal{Q}$ may not necessarily be integers. Due to the finite, discrete nature of $\mathcal{Q}$, solving (8) is NP-hard. In this paper, we call (8) an MILP no matter whether $\mathcal{Q}$ is an integer set. Our BnB algorithms developed in the following apply to both cases.

For the performance using the l_1 norm (3), we have the following theorem:

Theorem 1: The solution of the l_1 minimization problem (3) achieves the same diversity order as that of the MLD (2).

Proof is omitted for brevity. Theorem 1 indicates that if the noise follows the normal distribution, the conventional MLD only has an SNR gain over (3) and nothing more than that. From the simulation results, we find the SNR gain is also small. Therefore, (3) has almost the same performance as the conventional MLD, and the former is robust to impulsive noise. Moreover, LP technologies are mature and a multitude of implementations is available. All in all, (3) has several advantages over (2) from a practical point of view.

B. BnB algorithms

The most widely used method for solving MILP is BnB. Our first BnB algorithm is given in Algorithm 1. Eq. (9) can be solved using the simplex algorithm or the ellipsoid algorithm [9] in polynomial time.

BnB1: (**l**, **u**, UB)
output: $\check{\mathbf{x}}$, UB

1 Solve the LP

$$\begin{array}{lll} \min & \sum_{i=1}^{2n} t_i \\ \text{s.t.} & \mathbf{H}\mathbf{x} - \mathbf{t} \le \mathbf{y} \\ & -\mathbf{H}\mathbf{x} - \mathbf{t} \le -\mathbf{y} \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{array} \tag{9}$$

obtain the minimum v and $\hat{\mathbf{x}}$ achieving v ;
2 **if** *(9) is infeasible or* $v > UB$ **then return** ;
3 **if** $\hat{\mathbf{x}} \in \mathcal{Q}^{2n}$ **then**
4 | **if** $v < UB$ **then** $\check{\mathbf{x}} = \hat{\mathbf{x}}$, $UB = v$;
5 | **return**;
6 **end**
7 Find the first $\hat{x}_k$ such that $\hat{x}_k \notin \mathcal{Q}$;
8 Update the k-th entry of **u**, u_k to the largest element q in $\mathcal{Q}$ such that $q < \hat{x}_k$;
9 [$\mathbf{x}_1$, v_1] = BnB1(**l**, **u**, $\hat{\mathbf{x}}$, UB);
10 **if** $v_1 < UB$ **then** $\check{\mathbf{x}} = \mathbf{x}_1$, $UB = v_1$; **else** $\check{\mathbf{x}} = \hat{\mathbf{x}}$;
11 Update the k-th entry of **l**, l_k to the smallest element q in $\mathcal{Q}$ such that $q > \hat{x}_k$;
12 [$\mathbf{x}_1$, v_1] = BnB1(**l**, **u**, $\hat{\mathbf{x}}$, UB);
13 **if** $v_1 < UB$ **then** $\check{\mathbf{x}} = \mathbf{x}_1$, $UB = v_1$; **else** $\check{\mathbf{x}} = \hat{\mathbf{x}}$;

Algorithm 1: BnB algorithm 1 for MILP

Let $\rho_{\min}$ and $\rho_{\max}$ denote the minimum value and maximum value in $\mathcal{Q}$, respectively. Algorithm 1 is invoked as BnB1($\rho_{\min}\mathbf{1}, \rho_{\max}\mathbf{1}, \mathbf{0}, +\infty$), where $\mathbf{1}$ and $\mathbf{0}$ are all ones and zeros vectors, respectively. Algorithm 1 is a simple modification of the algorithm in [10]. We replace the branch process in [10] by updating $u_k = \lfloor \hat{x}_k \rfloor$ and $l_k = \lceil \hat{x}_k \rceil$ with lines 8

and 11 in Algorithm 1 by considering the discrete nature of $\mathcal{Q}$, where $\lfloor \hat{x}_k \rfloor$ ($\lceil \hat{x}_k \rceil$) denotes the largest (smallest) integer smaller (greater) than or equal to $\hat{x}_k$.

However, if Q has a very large size, the search tree formed by BnB1 will span many nodes. Moreover, if $\mathcal{Q}$ has infinite elements, BnB1 cannot work and it is trapped in an endless loop. We next give an improved BnB algorithm to avoid these problems, which is given in Algorithm 2.

In Algorithm 2, given the bound on $\sum_{i=1}^{2n} t_i$, we can find the lower bound of x_k by solving the LP (10), and similarly the upper bound of x_k. Even if $\mathcal{Q}$ has infinite elements, the set $\mathcal{Q} \cap [lb, ub]$ is finite and the algorithm terminates after a finite number of steps. Line 10 is introduced because we want to have a small candidate set for x_k in the first few stages. Note that in Algorithm 2 the constraint $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$ in all the LPs are not necessary. The price we pay in Algorithm 2 is using 2 additional LPs to get lb and ub. If the size of $\mathcal{Q}$ is very large, BnB2 saves complexity over BnB1.

BnB2: (i, $\mathbf{y}$, $\mathbf{H}$, UB)
output: $\check{\mathbf{x}}$, UB

1 **if** $i == 0$ **then return**;
2 Solve the LP (9), obtain the minimum v and $\hat{\mathbf{x}}$ achieving v ;
3 **if** *(9) is infeasible or* $v > UB$ **then return** ;
4 **if** $\hat{\mathbf{x}} \in \mathcal{Q}^{2n}$ **then**
5 | **if** $v < UB$ **then** $\check{\mathbf{x}} = \hat{\mathbf{x}}$, $UB = v$;
6 | **return**;
7 **end**
8 Map $\hat{\mathbf{x}}$ to $\mathcal{Q}$ and the resulting vector is $\mathbf{x}_0$, and compute $v_0 = \|\mathbf{y} - \mathbf{H}\mathbf{x}_0\|_1$;
9 **if** $v_0 < UB$ **then** $UB = v_0$, $\check{\mathbf{x}} = \mathbf{x}_0$;
10 Find $k = \arg\min_j \min_{q \in \mathcal{Q}} |\hat{x}_j - q|$;
11 Solve

$$\begin{aligned} \min \quad & x_k \\ \text{s.t.} \quad & \mathbf{H}\mathbf{x} - \mathbf{t} \le \mathbf{y} \\ & -\mathbf{H}\mathbf{x} - \mathbf{t} \le -\mathbf{y} \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u} \\ & \textstyle\sum_{i=1}^{2n} t_i \le UB, \end{aligned} \tag{10}$$

obtain the minimum lb;
12 Solve $\max x_k$ under the same constraints as (10), and obtain the maximum ub;
13 **for** $x_k \in \mathcal{Q} \cap [lb, ub]$ **do**
14 | $\mathbf{y}_1 = \mathbf{y} - \mathbf{H}(:, k)x_k$, delete the k-th column of $\mathbf{H}$;
15 | [$\mathbf{x}_1$, v_1] = BnB2($i-1$, $\mathbf{y}$, $\mathbf{H}$, UB);
16 | **if** $v_1 < UB$ **then** $\check{\mathbf{x}} = \mathbf{x}_1$, $UB = v_1$, and add x_k to the k-th entry of $\check{\mathbf{x}}$;
17 **end**

Algorithm 2: BnB algorithm 2 for MILP

Several improvements can be applied to reduce the complexity of the BnB algorithms. In Algorithm 1, when branching down from a node, some inequality constraints in $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$ may become equality constraints. We can write these equalities explicitly to reduce the number of constraints.

Simplex algorithm has two phases [9]. In phase I, the simplex algorithm solves another LP to find a vertex. The BnB algorithm may need to solve a large number of LPs to find the optimal solution. Therefore, it is important to reuse the results of existing LP solutions. The similarities between parent and child problems allow us to greatly simplify the first stage of the simplex algorithm. The parent and child problems differ by only one constraint. For example, if the lower bound l_k on x_k in the parent problem is changed to l'_k in the child problem (line 12 in Algorithm 1), we introduce a slack variable s_k and solve an LP given by

$$\begin{aligned} \min \quad & s_k \\ \text{s.t.} \quad & \mathbf{H}\mathbf{x} - \mathbf{t} \le \mathbf{y} \\ & -\mathbf{H}\mathbf{x} - \mathbf{t} \le -\mathbf{y} \\ & l_i \le x_i \le u_i, i = 1, \ldots, 2n,\ i \ne k \\ & l'_k \le x_k + s_k \le u_k,\ s_k \ge 0. \end{aligned} \tag{11}$$

If the solution to the parent problem is $[\hat{\mathbf{x}}, \hat{\mathbf{t}}]$, $[\hat{\mathbf{x}}, \hat{\mathbf{t}}, l_k - l'_k]$ is an obvious vertex of (11). If the minimum of (11) is 0, the optimal solution to (11) is a vertex of the child problem, and the phase I of the simplex algorithm for the child problem can start from this vertex. If the minimum of (11) is not zero, the child problem is infeasible. Typically, only a few pivots are required to solve (11), much fewer than the number needed by the phase I in standard simplex algorithm. This technique can also be applied to Algorithm 2. In Algorithm 2, the optimal solution to (9) is also feasible for the LPs in lines 11 and 12. Therefore, two LPs in phase I are eliminated.

C. LP relaxation

To reduce the complexity of the MILP, we can further relax the MILP (8) as an LP by removing the constellation constraint on x_i. The LP detector is given by

$$\begin{aligned} (\mathrm{LP}_1) \quad \min \quad & \textstyle\sum_{i=1}^{2n} t_i \\ \text{s.t.} \quad & \mathbf{H}\mathbf{x} - \mathbf{t} \le \mathbf{y} \\ & -\mathbf{H}\mathbf{x} - \mathbf{t} \le -\mathbf{y} \\ & \rho_{\min} \le x_i \le \rho_{\max},\ i = 1, \ldots, 2n. \end{aligned} \tag{12}$$

The LP (12) can be solved using the ellipsoid algorithm in polynomial time. If $\mathcal{Q}$ has only two elements (i.e., 4QAM) and the solution to (12) is $\hat{\mathbf{x}}$, we have the following theorem.

Theorem 2: If $\hat{\mathbf{x}} \in \mathcal{Q}^{2n}$, $\hat{\mathbf{x}}$ is guaranteed to be the solution of (8).

Proof: All the feasible points to (8) are also feasible to (12). If $\hat{\mathbf{x}} \in \mathcal{Q}^{2n}$, $\hat{\mathbf{x}}$ is feasible to (8), and it attains the optimum of (12). Therefore, it also attains the optimum of (8).

From the simulation results, we find the LPD (12) has good performance but it still has a large gap from (8) and MLD. To improve the performance of the LPD (12) with little increase in complexity, we propose to find the most unreliable symbol x_k first. For each $x_k \in \mathcal{Q}$, we cancel its contribution to $\mathbf{y}$, and use the LPD (12) to solve for the remaining $2n-1$ symbols. The resulting vectors are denoted by $\hat{\mathbf{x}}^i$, $i = 1, \ldots, |\mathcal{Q}|$. The vector attains the minimum of $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^i\|_1$ is output as the solution. The reliability of data symbols can be measured by solving (12) first, giving the optimal solution $\hat{\mathbf{x}}$. $k = \arg\max_j \min_{q \in \mathcal{Q}} |\hat{x}_j - q|$ is the

index of the most unreliable symbol. This approach can also be generalized to do ML search for the first K most unreliable symbols. This detector is denoted by list LPD or LLPD.

Remarks:

- No matrix decomposition is needed in both (8) and (12) as opposed to SD [2]. Therefore, our LPDs can be directly applied to rank deficient systems without modification.
- The simplex algorithm can be readily parallelized. The LPD (8) can be immediately implemented using mature VLSI chip design for simplex algorithm.
- Due to the inherent nature of BnB algorithm, our BnB algorithms for solving MILP also have small complexity in high SNR and high complexity in low SNR, which is similar to SD.

IV. Linear programming decoding for coded MIMO systems

In [11], two coding architectures, vertical coding and horizontal coding, are proposed for BLAST systems. In this paper, we consider the vertical coding with binary linear codes, where a binary linear code is used to encode all the data bits going to different antennas. Our decoder can be readily generalized to the horizontal coding architecture.

At the transmitter, k independent information symbols $\mathbf{b} \in \{0,1\}^k$ are first encoded via a binary linear code $\mathcal{C}$ with generator matrix $\mathbf{G}$ to produce a codeword $\mathbf{c} = \mathbf{G} \odot \mathbf{b} \in \{0,1\}^{nM_c}$, where $\odot$ denotes multiplication over $\mathrm{GF}(2)$. Every M_c bits in $\mathbf{c}$ are subsequently mapped to an element in $\mathcal{Q}$ with 2^{M_c} elements. The resulting modulated vector $\tilde{\mathbf{x}} = \mathcal{M}(\mathbf{c}) \in \mathcal{Q}^n$, where $\mathcal{M}(\cdot)$ denotes the mapping operation.

If $\tilde{x}_i$ is from M-QAM and Gray mapping is used, we can always transform a system with high order QAM to an equivalent system with 4QAM using the approach in [12]. Therefore, we only discuss 4QAM system in the following. Each bit in $\mathbf{c} \in \{0,1\}^{2n}$ is mapped to a symbol $\mathbf{x} \in \{-1,1\}^{2n}$ in (4). We assume the mapping $0 \to -1$ and $1 \to +1$, or equivalently $\mathbf{x} = 2\mathbf{c} - 1$. Using the l_1 norm metric, we can detect the information bits by

$$(\mathrm{P}_2) \quad \min_{\mathbf{c}\in\mathcal{C}} \|\mathbf{y} - \mathbf{H}(2\mathbf{c}-1)\|_1. \tag{13}$$

We first show how to relax (13) as an LP. For a given code $\mathcal{C}$, we define the codeword polytope to be the convex hull of all possible codewords

$$\mathrm{poly}(\mathcal{C}) = \left\{ \sum_{\mathbf{f}\in\mathcal{C}} \lambda_{\mathbf{f}} \mathbf{f} : \lambda_{\mathbf{f}} \geq 0, \sum_{\mathbf{f}\in\mathcal{C}} \lambda_{\mathbf{f}} = 1 \right\}. \tag{14}$$

$\mathrm{poly}(\mathcal{C})$ is a polytope contained within the hypercube $[0,1]^{2n}$ with its vertices corresponding to the codewords of $\mathcal{C}$. (P_2) can be relaxed as

$$(\mathrm{P}_3) \quad \min_{\mathbf{c}\in\mathrm{poly}(\mathcal{C})} \|\mathbf{y} - \mathbf{H}(2\mathbf{c}-1)\|_1. \tag{15}$$

(P_3) is equivalent to

$$\begin{aligned} (\mathrm{LP}_2) \quad \min \quad & \textstyle\sum_{i=1}^{2n} t_i \\ \text{s.t.} \quad & \mathbf{H}(2\mathbf{c}-1) - \mathbf{t} \leq \mathbf{y} \\ & -\mathbf{H}(2\mathbf{c}-1) - \mathbf{t} \leq -\mathbf{y} \\ & \mathbf{c} = \textstyle\sum_{\mathbf{f}\in\mathcal{C}} \lambda_{\mathbf{f}} \mathbf{f} \\ & \textstyle\sum_{\mathbf{f}\in\mathcal{C}} \lambda_{\mathbf{f}} = 1,\ \lambda_{\mathbf{f}} \geq 0, \end{aligned} \tag{16}$$

where $\mathbf{t} = [t_1, \ldots, t_{2n}] \in \mathbb{R}^{2n}$ and $\lambda_{\mathbf{f}}$ are the optimization variables. Although (P_3) is relaxed to (LP_2), the number of constraints in (LP_2) is exponential in the code length $2n$. Both simplex and ellipsoid algorithms have exponential complexity for solving (LP_2).

In [6], a relaxed polytope is formulated. For each row $\mathbf{h}_j$, $j = 1, \ldots, 2n-k$ of the parity-check matrix of $\mathcal{C}$, let $U(\mathbf{h}_j)$ be the support of $\mathbf{h}_j$, or the set of positions of 1 in $\mathbf{h}_j$. Each codeword $\mathbf{c} \in \mathcal{C}$ must satisfy the following inequalities [6]: for each set $V \subseteq U(\mathbf{h}_j)$ such that $|V|$ is odd,

$$\sum_{i\in V} c_i - \sum_{i\in U(\mathbf{h}_j)\setminus V} c_i \leq |V| - 1. \tag{17}$$

Therefore, we have the LP joint decoder as

$$\begin{aligned} (\mathrm{LP}_3) \quad \min \quad & \textstyle\sum_{i=1}^{2n} t_i \\ \text{s.t.} \quad & \mathbf{H}(2\mathbf{c}-1) - \mathbf{t} \leq \mathbf{y} \\ & -\mathbf{H}(2\mathbf{c}-1) - \mathbf{t} \leq -\mathbf{y} \\ & \displaystyle\sum_{i\in V} c_i - \sum_{i\in U(\mathbf{h}_j)\setminus V} c_i \leq |V| - 1 \\ & 0 \leq c_i \leq 1,\ i = 1, \ldots, 2n. \end{aligned} \tag{18}$$

From Theorem 2 and Proposition 2 in [6], if $\hat{\mathbf{c}}$ is the solution to (18), we have the following theorem.

Theorem 3: If $\hat{\mathbf{c}} \in \{0,1\}^{2n}$, $\hat{\mathbf{c}}$ is guaranteed to be the solution of (13).

If $\hat{\mathbf{c}} \notin \{0,1\}^{2n}$, we make hard decisions on $\hat{\mathbf{c}}$, and recover the transmitted bits $\mathbf{b}$ with a syndrome decoder. Or we can do soft-decision decoding on $\hat{\mathbf{c}}$ directly.

Using Theorem 3, we can solve (13) exactly as an MILP by allowing integer constraints on c_i. (P_2) is equivalent to

$$\begin{aligned} (\mathrm{MILP}_2) \quad \min \quad & \textstyle\sum_{i=1}^{2n} t_i \\ \text{s.t.} \quad & \mathbf{H}(2\mathbf{c}-1) - \mathbf{t} \leq \mathbf{y} \\ & -\mathbf{H}(2\mathbf{c}-1) - \mathbf{t} \leq -\mathbf{y} \\ & \displaystyle\sum_{i\in V} c_i - \sum_{i\in U(\mathbf{h}_j)\setminus V} c_i \leq |V| - 1 \\ & c_i \in \{0,1\},\ i = 1, \ldots, 2n. \end{aligned} \tag{19}$$

(MILP_2) can be solved using the BnB algorithms 1 and 2.

Note that the approach proposed in this paper can also be extended to decoding coded MIMO systems with turbo-like codes by considering the results in [13], which may achieve near-capacity on a MIMO system.

V. Simulation results

We now present the error rates of our proposed linear programming detectors for a MIMO system over a flat Rayleigh fading channel.

Fig. 1 shows the symbol error rate (SER) of different linear programming detectors in a 8×8 MIMO system with 4QAM.

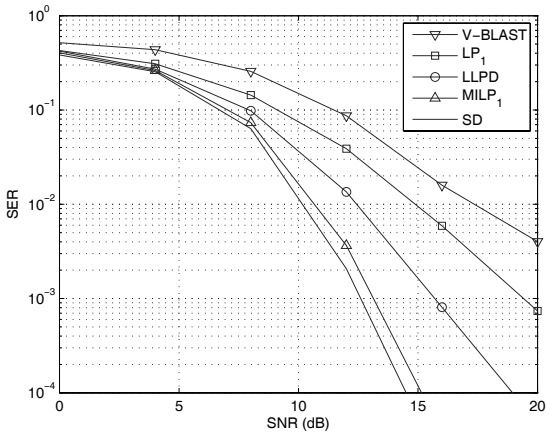

Fig. 1. Performance comparison of linear programming detectors in an 8×8 MIMO system with 4OAM.

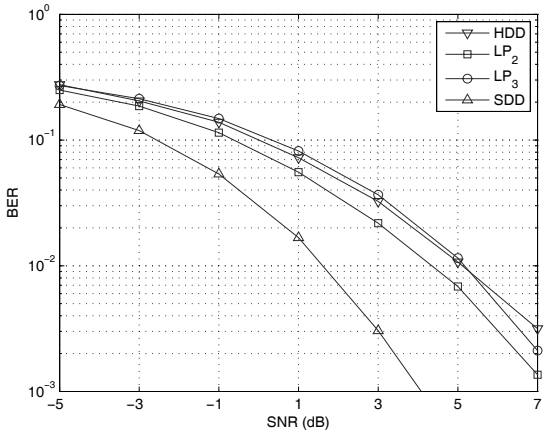

Fig. 2. Performance comparison of different detectors in a $(8, 4)$ extended Hamming coded 8×8 MIMO system with BPSK.

All our proposed LPDs outperform V-BLAST. Even LP_1 has a 2.5-dB gain over V-BLAST at SER$= 10^{-2}$. More importantly, it appears that the diversity order of LP_1 is 2, but V-BLAST only has diversity order 1. Note that LP_1 is also a polynomial time detector. With simple modifications, LLPD has a 3.7-dB gain over LP_1 at SER$= 10^{-3}$. MILP_1 performs close to SD. The performance gap between these two is only 0.5 dB at SER$= 10^{-4}$. We also find MILP_1 achieves the same diversity order as MLD, which is consistent with Theorem 1.

In Fig. 2, we compare the bit error rate (BER) of different decoders in a $(8, 4)$ extended Hamming coded 8×8 MIMO system with BPSK. Hard decision decoding and soft decision decoding are denoted as HDD and SDD, respectively. We choose this simple example because the SDD can be performed by exhaustive search, and we can learn the performance loss by using our linear programming decoders. LP_3 achieves almost the same performance as HDD. However, LP_3 does not need to detect the coded symbols first as HDD does, which may have high complexity. LP_2 performs better than both HDD and LP_3. At BER$= 10^{-2}$, LP_2 has a 0.8-dB gain over LP_3. But the number of constraints in LP_2 is exponential in $2n$, which is computationally inefficient. Both LP_2 and LP_3 perform inferior to SDD. At BER$= 2 \times 10^{-3}$, SDD has a 3.1-dB gain over LP_2. This performance gap motivates the search for alternative tight relaxations.

VI. Conclusion

In this paper, we have given several linear programming detectors and decoders for MIMO systems. All the proposed detectors are based on the metric formulated by l_1 norm as opposed to the l_2 norm used in conventional detectors. The l_1 minimization problem is first transformed into an MILP. We have proposed two Branch-and-Bound algorithms to solve the resulting MILP, the second of which applies to infinite integer sets. Several improvements were also discussed for the BnB algorithms. The MILP is further relaxed to an LP, which can be solved in polynomial time. For decoding in binary linear coded MIMO systems, the information bits are relaxed to lie in a codeword polytope instead of taking only values $\{0, 1\}$, and an LP decoder is proposed. However, this LP has an exponential number of constraints. A relaxed polytope is then formulated, which has the property that all the vertices correspond to codewords. Exact decoding can also be attained by solving an MILP after imposing an integer constraint on the information bits. Interesting future work includes analyzing the performance of different linear programming detectors for uncoded systems, and deriving the performance bounds of linear programming decoders for coded systems.

References

[1] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, no. 1, pp. 14–15, Jan. 1999.

[2] E. Viterbo and J. Bouros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

[3] A. Burg, M. Borgmann, C. Simon, M. Wenk, M. Zellweger, and W. Fichtner, "Performance tradeoffs in the VLSI implementation of the sphere decoding algorithm," in *Proc. IEE 3G Mobile Communication Technologies Conference*, London, UK, Oct. 2004.

[4] E. J. Candes and T. Tao, "Error correction via linear programming," in *Proc. of IEEE Symposium on Foundations of Computer Science*, 2005.

[5] D. L. Donoho, "For most large underdetermined systems of equations, the minimal l_1-norm near-solution approximates the sparsest near-solution," *Commun. on Pure and Applied Math.*, vol. 59, no. 7, pp. 907 – 934, Mar. 2006.

[6] J. Feldman, M. Wainwright, and D. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 954 – 972, March 2005.

[7] A. Bertossi and M. Bonuccelli, "A VSLI implementation of the simplex algorithm," *IEEE Trans. Comput.*, vol. 36, no. 2, pp. 241 – 247, Feb. 1987.

[8] W. Li and J. Swetits, "The linear l_1 estimator and the huber m-estimator," *SIAM J. Optimization*, vol. 8, pp. 457– 475, 1998.

[9] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[10] R. Dakin, "A tree-search algorithm for mixed integer programming problems," *The Computer Journal*, vol. 8, no. 3, pp. 250–255, 1965.

[11] X. Li, H. Huang, G. Foschini, and R. Valenzuela, "Effects of iterative detection and decoding on the performance of BLAST," in *Proc. of IEEE GLOBECOM*, vol. 2, Nov 2000, pp. 1061 – 1066.

[12] T. Cui and C. Tellambura, "An efficient generalized sphere decoder for rank-deficient MIMO systems," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 423 – 425, May 2005.

[13] J. Feldman and D. R. Karger, "Decoding turbo-like codes via linear programming," *Journal of Computer and System Sciences*, vol. 68, no. 4, pp. 733–752, June 2004.